In the claims

1. (currently amended) A method comprising:

assigning interrupts for a plurality of input/output (I/O) devices among a plurality of nodes of a system based on at least one of: the nodes to which the I/O devices are connected; the nodes at which interrupt service routines for the I/O devices reside; and, processors of the nodes for the nodes having processors, where one or more of the nodes are processorless and/or memoryless and one or more other of the nodes have processors and memory,

wherein assigning the interrupts for the I/O devices among the nodes of the system comprises, for a given interrupt for a given I/O device, the given I/O device connected to a first node of the nodes of the system, the given I/O device having an interrupt service routine to handle the given interrupt, the interrupt service routine residing at a second node of the nodes of the system, the first node being different than the second node, in order from (a)-(c):

(a) where [[if]] the first node to which the given I/O device is connected has a cache, memory, and at least one processor, and the second node at which the interrupt service routine of the given I/O device resides does have a cache, memory, and at least one processor, ~~then~~ assigning the given interrupt for the given I/O device to the first node~~, even if the second node has a cache, memory, and at least one processor~~;

(b) where [[if]] the first node does not have a cache, memory, and at least one processor, [[but]] the second node at which the interrupt service routine of the given I/O device resides does have a cache, memory, and at least one processor, and a third node of the nodes of the nodes of the system has memory and at least one processor, ~~then~~ assigning the given interrupt for the given I/O device to the second node~~, even if a third node has memory and at least one processor~~;

(c) where [[if]] both the first node and the second node do not each have a cache, memory, and at least one processor, ~~then~~ assigning the given interrupt for the given I/O device to the third node ~~of the nodes of the system, the third node having memory and at least one processor~~, the third node being different than the first node and the second node;

for each node having processors, assigning the interrupts for the devices that are performance critical and that have been assigned to the node among the processors of the node in a round-robin manner;

dynamically modifying assignments of the interrupts among the nodes of the system based on actual performance characteristics of the assignments; and,

for each node having processors, dynamically modifying assignments of the interrupts that are performance critical and that have been assigned to the node among the processors of the node based on actual performance characteristics of the assignments.

2. (cancelled)

3. (cancelled)

4. (previously presented) The method of claim 1, wherein dynamically modifying the assignments of the interrupts among the nodes of the system comprises, for each assignment of an interrupt for an I/O device to a node, where the node is that to which the I/O device is connected:

measuring responsiveness of the node in processing the interrupt;

assigning the interrupt to the node at which the interrupt service routine for the I/O device resides;

measuring responsiveness of the node at which the interrupt service routine for the I/O device resides in processing the interrupt; and, where the responsiveness of the node to which the I/O device is connected is better than the responsiveness of the node at which the interrupt service routine for the I/O device resides, reassigning the interrupt to the node to which the I/O device is connected.

5. (previously presented) The method of claim 4, wherein dynamically modifying the assignments of the interrupts among the nodes of the system comprises, for each assignment of an interrupt for an I/O device to a node, where the node is that at which the interrupt service routine for the I/O device resides:

measuring responsiveness of the node in processing the interrupt;

assigning the interrupt to the node to which the I/O device is connected;

measuring responsiveness of the node to which the I/O device is connected in processing the interrupt; and,

where the responsiveness of the node at which the interrupt service routine for the I/O device resides is better than the responsiveness of the node to which the I/O device is connected, reassigning the interrupt to the node at which the interrupt service routine for the I/O device resides.

6. (previously presented) The method of claim 1, wherein, for each node of the system_ having processors, dynamically modifying the assignments of the interrupts that are performance critical and that have been assigned to the node among the processors of the node comprises:

measuring responsiveness of the processors of the node in processing the interrupts assigned thereto;

where a differential between a best responsiveness and a worst responsiveness is greater than a threshold,

reassigning at least one of the interrupts assigned to the processor having the worst responsiveness to the processor having the best responsiveness.

7.-24. (cancelled)

25. (currently amended) A method comprising:

assigning interrupts for a plurality of input/output (I/O) devices among a plurality of nodes based on at least one factor selected from the set consisting of: the nodes to which the I/O devices are connected; and the nodes at which interrupt service routines for the I/O devices reside, where one or more of the nodes are processorless and/or memoryless and one or more other of the nodes have processors and memory,

wherein assigning the interrupts for the I/O devices among the nodes of the system comprises, for a given interrupt for a given I/O device, the given I/O device connected to a first node of the nodes of the system, the given I/O device having an interrupt service routine to handle the given interrupt, the interrupt service routine residing at a second node of the nodes of the system, the first node being different than the second node, in order from (a)-(c):

(a) where [[if]] the first node to which the given I/O device is connected has a cache, memory, and at least one processor, and the second node at which the interrupt service routine of the given I/O device resides does have a cache, memory, and at least one processor, ~~then~~ assigning the given interrupt for the given I/O device to the first node~~, even if the second node has a cache, memory, and at least one processor~~;

(b) where [[if]] the first node does not have a cache, memory, and at least one processor, [[but]] the second node at which the interrupt service routine of the given I/O device resides does have a cache, memory, and at least one processor, and a third node of the nodes of the nodes of the system has memory and at least one processor, ~~then~~ assigning the given interrupt for the given I/O device to the second node~~, even if a third node has memory and at least one processor~~;

(c) where [[if]] both the first node and the second node do not each have a cache, memory, and at least one processor, ~~then~~ assigning the given interrupt for the given I/O device to the third node~~ of the nodes of the system, the third node having memory and at least one processor~~, the third node being different than the first node and the second node;

for each node of the system having processors, assigning the interrupts for the devices that are performance critical and that have been assigned to the node among the processors of the node in a round-robin manner;

dynamically modifying assignments of the interrupts among the nodes of the system based on actual performance characteristics of the assignments; and,

for each node of the system having processors, dynamically modifying assignments of the interrupts that are performance critical and that have been assigned to the node among the processors of the node based on actual performance characteristics of the assignments.

26. (cancelled)

27. (cancelled)

28. (original) The method of claim 25, wherein dynamically modifying the assignments of the interrupts among the nodes of the system comprises, for each assignment of an interrupt for an I/O device to a node, where the node is that to which the I/O device is connected:

measuring responsiveness of the node in processing the interrupt;

assigning the interrupt to the node at which the interrupt service routine for the I/O device resides;

measuring responsiveness of the node at which the interrupt service routine for the I/O device resides in processing the interrupt; and, where the responsiveness of the node to which the I/O device is connected is better than the responsiveness of the node at which the interrupt service routine for the I/O device resides, reassigning the interrupt to the node to which the I/O device is connected.

29. (original) The method of claim 28, wherein dynamically modifying the assignments of the interrupts among the nodes of the system comprises, for each assignment of an interrupt for an I/O device to a node, where the node is that at which the interrupt service routine for the I/O device resides:

measuring responsiveness of the node in processing the interrupt;

assigning the interrupt to the node to which the I/O device is connected;

measuring responsiveness of the node to which the I/O device is connected in processing the interrupt; and,

where the responsiveness of the node at which the interrupt service routine for the I/O device resides is better than the responsiveness of the node to which the I/O device is connected, reassigning the interrupt to the node at which the interrupt service routine for the I/O device resides.

30. (previously presented) The method of claim 25, wherein, for each node of the system having memory, dynamically modifying the assignments of the interrupts that are performance critical and that have been assigned to the node among the processors of the node comprises:

measuring responsiveness of the processors of the node in processing the interrupts assigned thereto;

where a differential between a best responsiveness and a worst responsiveness is greater than a threshold,

reassigning at least one of the interrupts assigned to the processor having the worst responsiveness to the processor having the best responsiveness.